



Doc Code: AP.PRE.REQ

PTO/SB/33 (07-05)

Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number (Optional)

50277-1766

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)]

on November 14, 2005

Signature

Typed or printed name Jennifer Newell

Application Number

10/040,578

Filed

12/28/01

First Named Inventor

David J. Long

Art Unit

2192

Examiner

Pham, Chrystine

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

- ☐ applicant/inventor.
- ☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.
(Form PTO/SB/96)

☒ attorney or agent of record.
Registration number _____

☐ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____

Signature

Stephen J. Shaw

Typed or printed name

(408) 414-1231

Telephone number

November 14, 2005

Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required.
Submit multiple forms if more than one signature is required, see below*.

☐ *Total of _____ forms are submitted.

This collection of information is required by 35 U.S.C. 132. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11, 1.14 and 41.6. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



REMARKS

As will be seen from the discussion below, there are clear errors of fact and law in the Examiner's rejections, as well as the Examiner's rejections being clearly improper and without basis.

Claim 1

Claim 1 requires, among other features, "associating with said instance of said class **an attribute that is not in said class or any superclass of said class.**"

In support of the §102(e) rejection based upon *Ng. et al.* (hereinafter "*Ng*") anticipating the above-referenced feature, the Examiner relies solely upon the assertion that the data member "*collection Orders_for_Customer*" of "Class Customer 420" in Fig. 4B is an "attribute that is **not in** said class [the "Customer" class] or any superclass of said class." This is clear error and requires no analysis or interpretation of the reference.

The "Customer" class described in *Ng* and relied upon by the Examiner is illustrated thusly in Figure 4B:

```
Class Customer {  
    int Cust_id;  
    str SSN;  
    collection Orders for Customer;  
    int      get_Cust_id();  
    void     set_Cust_id(int Cust_id);  
    str      get_SSN();  
    void     set_SSN(str SSN);  
    iterator getOrdersForCustomer();  
    void     addOrdersForCustomer(Order O);  
    void     removeOrdersForCustomer(Order O);  
}
```

(*Ng*, Fig. 4B) (emphasis added)

As is now obvious, the data member (*i.e.*, attribute) "*collection Orders_for_Customer*" clearly is **an attribute that IS in said class** (Customer), just as "*int Cust_id*" and "*str SSN*" are attributes of the Customer class.



Ng makes this clear:

Class 420 reflects customer table 202 and class 424 reflects order table 204. As such, class 420 contains a data member for customer ID, social security number, and a collection of objects representing the orders associated with that particular customer, thus implementing the foreign key. Class 420 also contains a number of methods to both get and set the value of the data members, including an iterator method to iterate through the order for this particular customer.

...

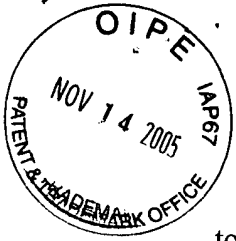
To define this relationship in the Java TM programming language, the class representing the referring table is defined to have a member that is a collection of the class representing the referred table. A "collection" refers to a type indicating a grouping of instances of other classes. Then, in the class reflecting the referred table, a member is added providing a reference to the class that refers to it.

(Ng, col. 6, lines 37-66)

Because the data member (*i.e.*, attribute) "*collection Orders_for_Customer*" clearly IS in the Customer class, the reference in no way anticipates the feature at issue, which requires "associating with said instance of said class an attribute that is not in said class or any superclass of said class."

"A claim is anticipated under § 102 only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

There is no "associating with said instance of said class an attribute that is not in said class or any superclass of said class" in the reference that could conceivably anticipate or in any way suggest the act of "associating with said instance of said class an attribute that is not in said class or any superclass of said class."



Claim 17

Claim 17 contains features that are similar to those described above with respect to Claim 1, and in particular both feature “associating with said instance of said class an attribute that is not in said class or any superclass of said class.”

Therefore, based on at least the reasons stated above with respect to Claim 1, there are clear errors of fact and law in the Examiner’s rejection of Claim 17, as well as the Examiner’s rejection being clearly improper and without basis.

Claims 16 and 32

Claims 16 and 32 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over *Ng* in view of The Java Virtual Machine Specification. Because the rejection incorporates the rejection of base Claim 1, and the rejection of Claim 1 has been previously demonstrated to be without merit, Applicant requests that the rejection of Claims 16 and 32 be reversed for the reasoning given above with respect to Independent Claims 1 and 17.

Remaining Dependent Claims

The pending claims not discussed so far are dependent claims that depend on an independent claim that is discussed above. Because each of the dependent claims includes the limitations of claims upon which it depends, the dependent claims are patentable for at least those reasons that the claims upon which the dependent claims depend are patentable.

Conclusion

Applicant requests that the rejections of all the pending claims be reversed.